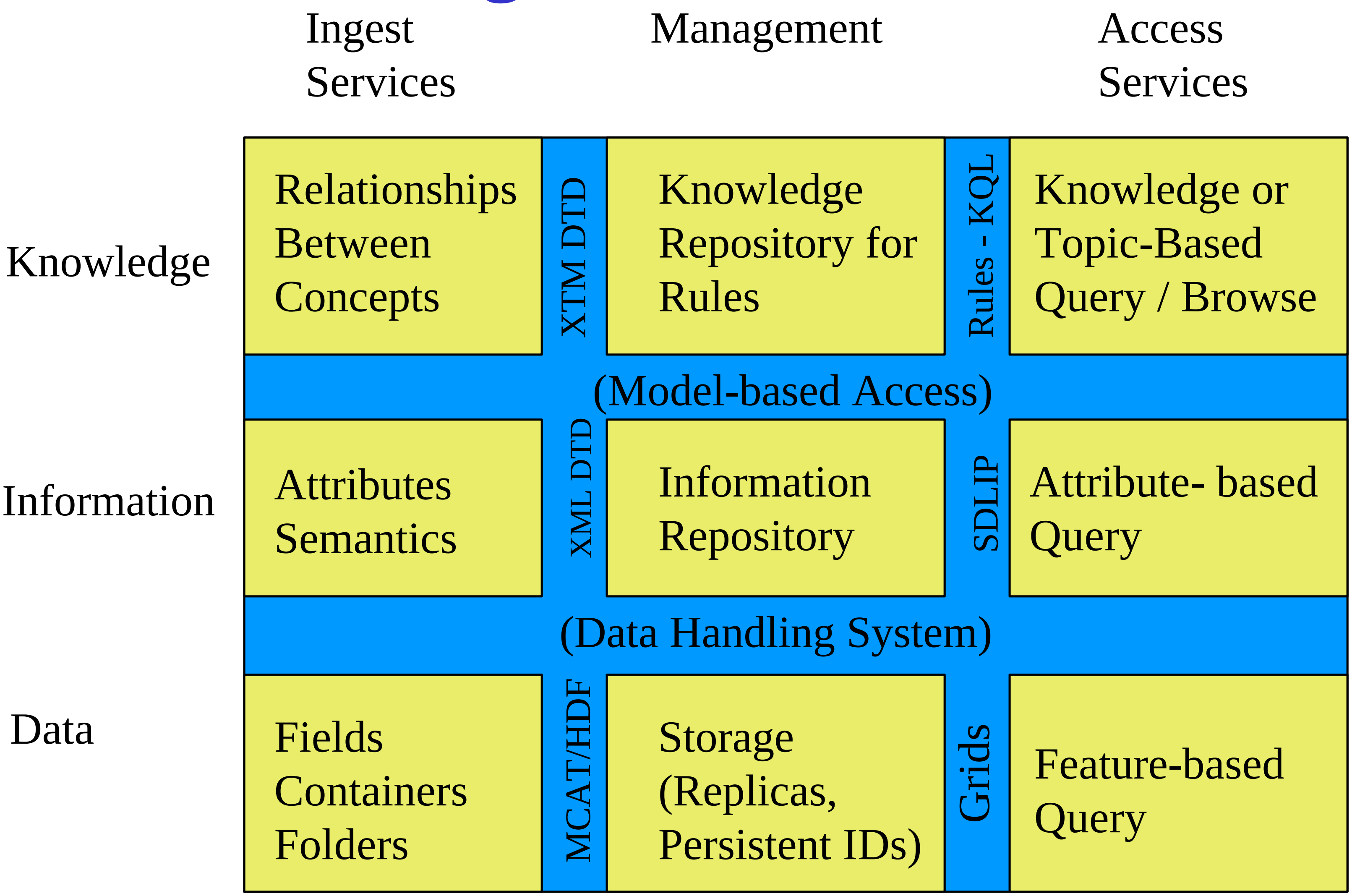
Knowledge-Based Grids: Two Use Cases

Reagan W. Moore
San Diego Supercomputer Center
9500 Gilman Drive, La Jolla, CA 92093-0505
Phone: 858 534-5073   FAX: 858 534-5152
E-mail: moore@sdsc.edu
http://www.npaci.edu/DICE/

Knowledge-Based Grid Architecture

- Based on the need to manage
  - Data objects
  - Information about the data objects (attributes)
  - Knowledge about the data objects (relationships)
- A “Virtual Data Grid” is an example of a Knowledge-based Grid.

Knowledge-Based Data Grids



	Ingest Services		Management		Access Services
Knowledge	Relationships Between Concepts	XTM DTD	Knowledge Repository for Rules	Rules - KQL	Knowledge or Topic-Based Query / Browse
			(Model-based Access)		
Information	Attributes Semantics	XML DTD	Information Repository	SDLIP	Attribute- based Query
			(Data Handling System)		
Data	Fields Containers Folders	MCAT/HDF	Storage (Replicas, Persistent IDs)	Grids	Feature-based Query

Use Cases

- NIH Biomedical Informatics Research Network
  - Federation of multiple existing digital libraries
  - Support information discovery, data access, data movement, and data analysis on distributed resources
- NARA Persistent Archive
  - Build a data collection that maintains authenticity of digital data while technology evolves
  - Support information discovery, data access, and migration to new data encoding standards

Queries across data sources from a common interface

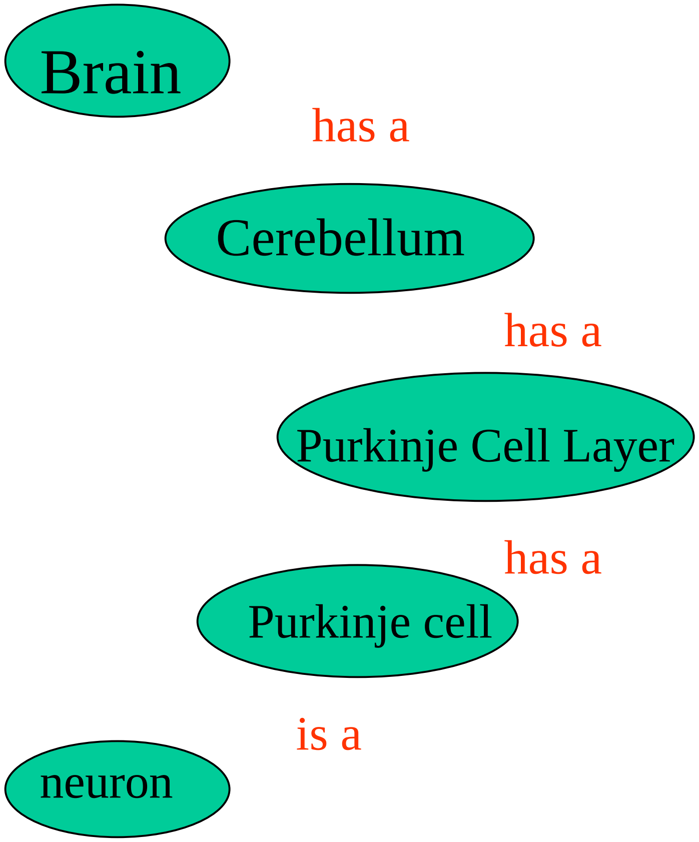
KIND Mediator

"How does the ***parallel fiber*** *output relate to the distribution of Ryanodine Receptors?"*

Sources: NCMIR UCSD / Yale Senselab

```
@SENSELAB: X1 := select output from parallel fiber ;
@MEDIATOR: X2 := "hang off" X1 from Domain Map;
@MEDIATOR: X3 := subregion-closure(X2);
@NCMIR:        X4 := select PROT-data(X3, Ryanodine Receptors);
@MEDIATOR: X5 := compute aggregate(X4);
```

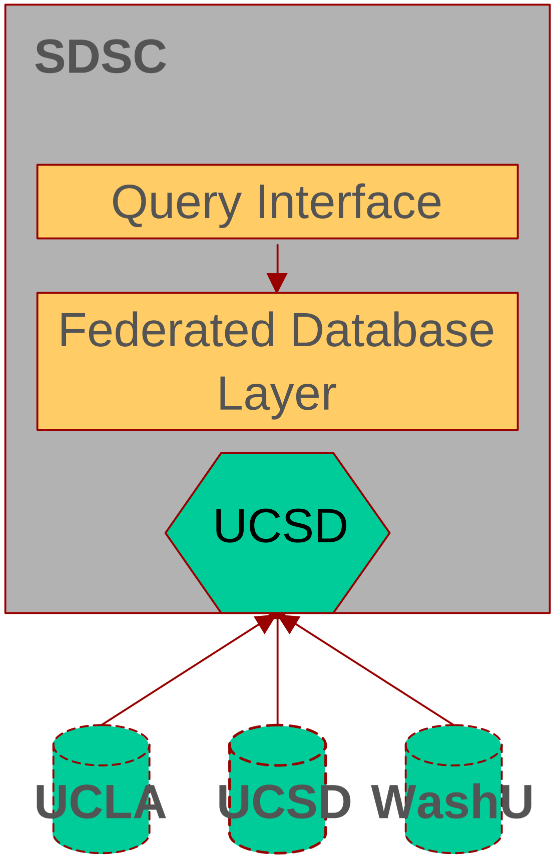
Use of Domain Maps to Navigate Data Sources



- Rule-based ontology maps
- Encodes conceptual and semantic relationships using F-logic

Federating Brain Data

Current Version



SDSC

Query Interface

Federated Database Layer

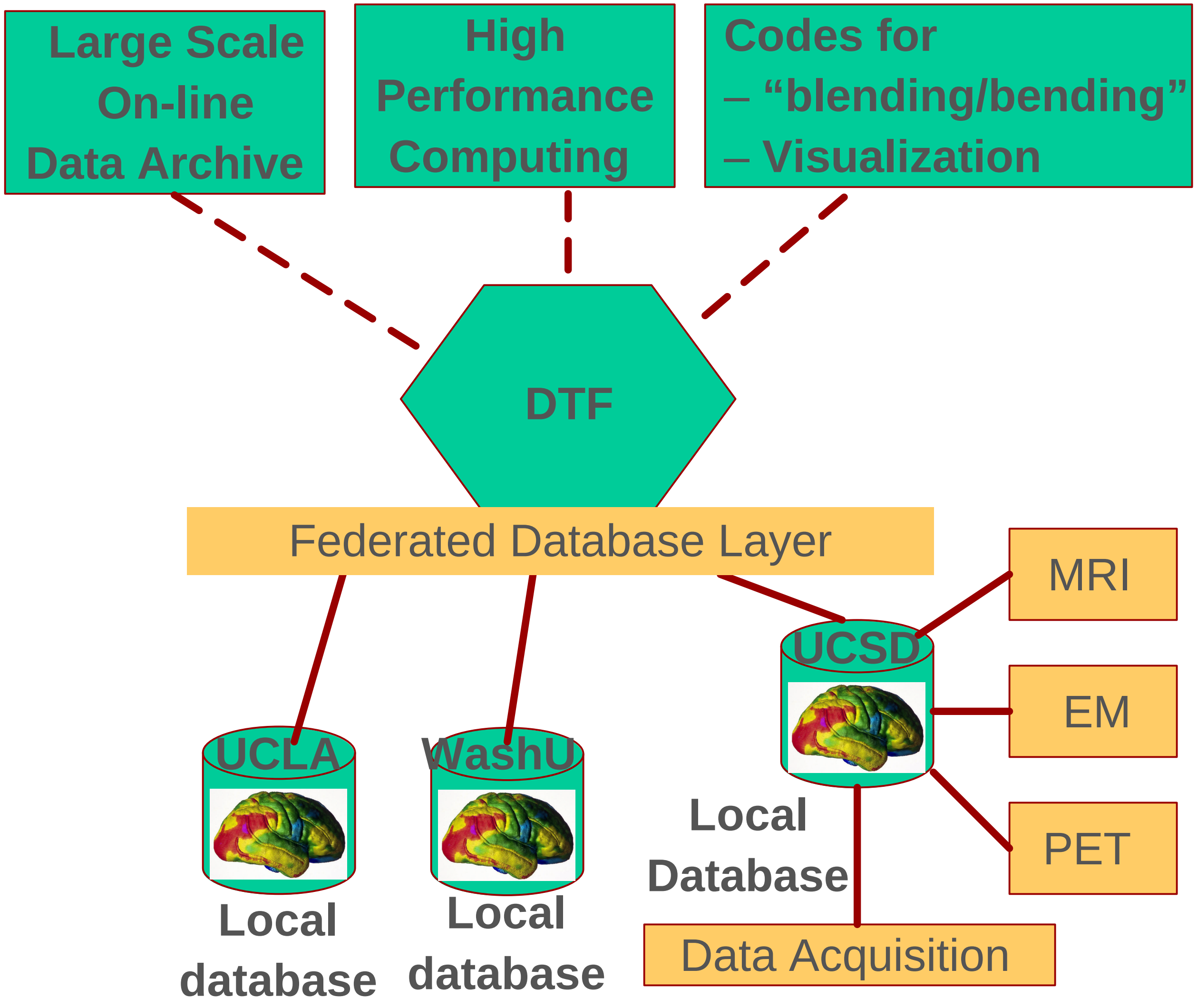
UCSD

UCLA

UCSD

WashU

With Distributed Terascale Facility



Large Scale On-line Data Archive

High Performance Computing

Codes for
– "blending/bending"
– Visualization

DTF

Federated Database Layer

UCLA

Local database

WashU

Local database

UCSD

Local Database

MRI

EM

PET

Data Acquisition

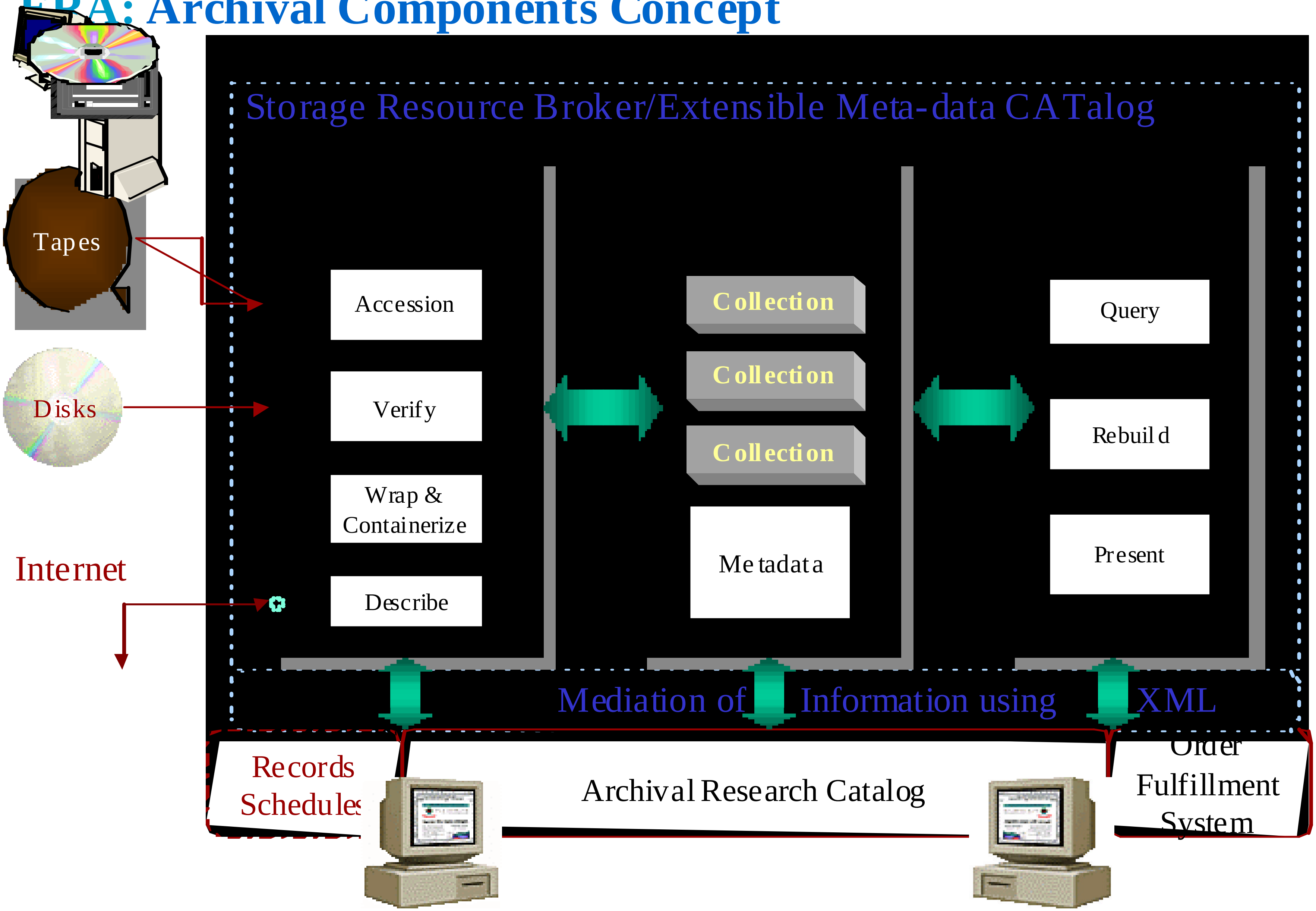
Grid Technology

- Model-based Mediation (Prolog system) to manage concept space
- Grid Portal to control telemicroscope
- Globus execution environment to analyze telemicroscope data
- MCAT Metadata Catalog to build union collection catalog
- Storage Resource Broker to manage access to collections and storage systems

Persistent Archive

- Manages data stored in an archive
- Uses collection to organize data that is being archived
- Uses an established encoding format for the data and for the collection
- Requires replication across physically distributed sites

ERA: Archival Components Concept



Storage Resource Broker/Extensible Meta-data CATalog

Tapes

Disks

Internet

Accession

Verify

Wrap & Containerize

Describe

Collection

Collection

Collection

Metadata

Query

Rebuild

Present

Mediation of Information using XML

Records Schedules

Archival Research Catalog

Order Fulfillment System

Knowledge-Based Data Grids

	Ingest Services		Management		Access Services
Knowledge	Relationships Between Concepts	XTM DTD	Knowledge Repository for Rules	Rules - KQL	Knowledge or Topic-Based Query / Browse
			(Model-based Access)		
Information	Attributes Semantics	XML DTD	Information Repository	SDLIP	Attribute- based Query
			(Data Handling System)		
Data	Fields Containers Folders	MCAT/HDF	Storage (Replicas, Persistent IDs)	Grids	Feature-based Query

Persistent Archive as a Data Grid

- Data grids provide most of the technology needed to create a persistent archive
  - Interoperation across heterogeneous storage systems
- Data grid federation in space is equivalent to persistent archive migration onto new technology
  - Both system need to simultaneously access heterogeneous storage systems

Virtual Data Concept

- Identify processes required to create a derived data product
- Provide collection management to organize derived data products
- Develop mechanism to create the derived data product if it is not available
- Requires management of relationships between derivation processes, input files, and output files

Virtual Data as basis for a Persistent Archive

- Dynamic migration of arbitrarily old data formats to the current encoding format used by current applications
- Dynamic migration of collection attributes to current information repository technology

- A persistent archive is a virtual data grid

Data Grid Requirements

- Support ownership of the data by the persistent archive
  - Requires management of access control lists independently of the storage system
- Require all data movement to be done through the persistent archive infrastructure
  - Integrated metadata update and data movement
  - Audit trails of all data accesses
- Provide metadata to track data integrity
  - Checksums, data signing

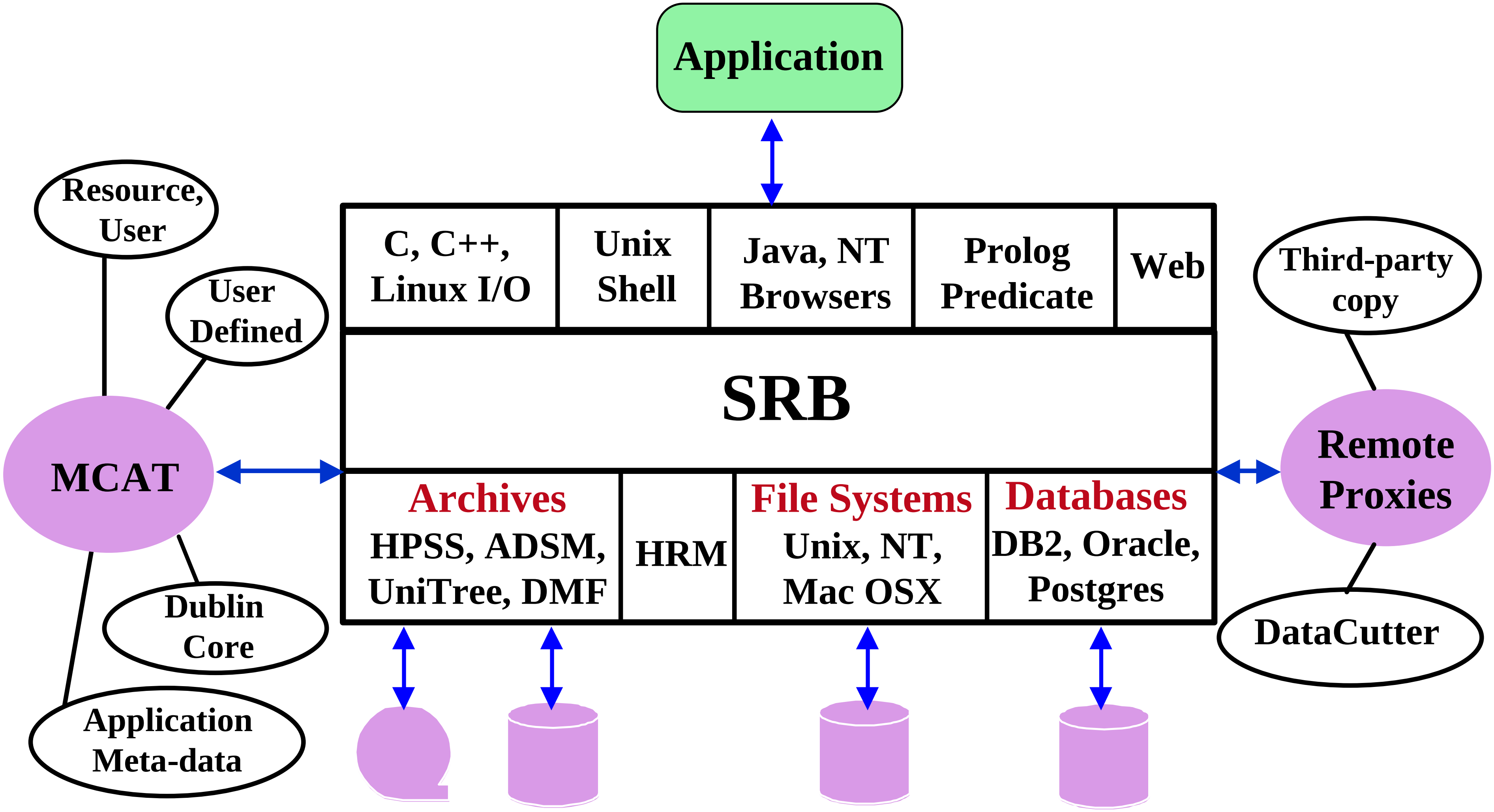
SDSC Storage Resource Broker

- Storage repository abstraction
  - Full Unix file system metadata and data operations
  - Client / Server architecture for adding drivers for new systems
- Information repository abstraction
  - Characterization of both the physical (table) and logical (schema) structures
  - Ability to migrate collection into new table structure on new information repository

Storage Resource Broker

- Logical name space
  - Replicas
  - Collection owned data
- Containers for aggregating data
  - Replicate containers
  - OAIS model for encapsulating data and metadata
- Collection for managing metadata
  - Export metadata as XML file
- Collection specific metadata
  - Audit trails

SDSC Storage Resource Broker & Meta-data Catalog



Grid Architecture

- Provide levels of abstraction for
  - Digital Entities
    - Data / Information / Knowledge
  - Repositories
    - Data / Information / Knowledge
  - Handling systems
    - Data / Information / Knowledge

Data Storage Abstraction - Data Operations

- Legion Persistent object semantics (get, put)
- Condor semantics (open, close, read, write)
- GridFTP protocol capabilities (get, put, open, close, read, write)
- SRB Unix file system operation support (open, close, read, write, seek, …)
- SRB Unix file system directory manipulation support (ls, dir, mkdir, …)

Data Storage Abstraction - Access

- Mappings to storage system logical structure
  - Direct links from the data grid name space to local files
  - Logical storage resource description that can represent multiple physical storage systems.
  - Fault tolerant logical storage resource description, write to “k” of “n” storage systems.
  - Shadow links from the data grid name space to directories in the local storage system.
- Data access abstraction implementation
  - Operating system I/O driver interface
  - Client – server architecture
  - Federated client – server architecture

Data Storage Abstraction - Control

- Data ownership – the local user ID under which the data is kept
  - Researcher owned data / Collection owned data / Grid owned data
- Authentication mechanism
  - Inter-realm authentication
  - Mapping to local authentication system via GSSAPI
- Access control mechanisms
  - Access control lists per data entity for each user / group
- Data granularity abstraction
  - Physical aggregation of files in containers.
    - Container locking
    - Container caching on disk when data is accessed in an archive
    - Container synchronization between disk cache and archive
  - Logical aggregation of files
    - Flat folder structure / Hierarchical folder structure
    - Soft links between folders to allow a file to be represented in multiple logical folders

Information Repository Abstraction

- Information management abstractions
  - Physical table structure
  - Schema
- Information access abstraction
  - Repository query mechanisms
  - Information discovery API
  - Attribute extraction mechanisms

Knowledge Repository Abstraction

- Knowledge management characterization
  - Organization
    - Concepts, semantic web, ontology
  - Mappings
    - Buckets, tokens, graphical
- Knowledge access characterization
  - Portal / Mediator / Logic spaces

Data Abstractions

- Encoding format
- Data representations
  - Replicas, versions, containers
- Data naming
  - Global name space
  - Logical name space
    - Organization - folders, hierarchical, soft links
    - Extensions - attributes
    - Consistency
    - Authenticity

Information Abstractions

- Representation syntax
- Aggregation syntax
- Transmission syntax
- Access control

Knowledge Abstractions

- Representation syntax
- Aggregation syntax
- Transmission syntax
- Access control

Data Handling Abstractions

- Latency management
- Transport
- Access API
  - C I/O library / C++ I/O library / Linux I/O redirection / Solaris I/O redirection / Java interface / Shell command interface / Web CGI interface / Windows browser interface / Predicate assertion interface
- Sharing controls

More Information

http://www.npaci.edu/DICE/